REMARKS

This Amendment is filed in response to the Office Action mailed on November 3, 2006. All objections and rejections are respectfully traversed.

Claims 1 to 12 and 14 to 51 are currently pending.

Claims 45-53 are added.

Request for Interview

The Applicant respectfully requests a telephonic interview with the Examiner after the Examiner has had an opportunity to consider this Amendment, but before the issuance of the next Office Action. The Applicant may be reached at 617-951-3067.

Claim Rejections – 35 USC § 112

At paragraph 3 of the Office Action, claims 1-12 and 14-44 are rejected under 35 U.S.C. §112, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Specifically, the Examiner states that it is not clear whether the allocation of at least one block per stripe to store parity occurs before or after writing of data in the stripe.

Applicant respectfully urges that representative claim 1 is clear in describing the order of the operations. The allocation of the redundant information is determined prior to writing the data and redundant information to the disks. The redundant information is assigned (allocated) to one of the unallocated blocks in the layout of the data blocks for the stripe. The redundant information is calculated based on the layout (placement) of data blocks on each disk of the stripe. Then the data and redundant information are written to disk according to the layout, which has the data allocated to certain blocks and the redundant information assigned to at least one block.

In representative claim 36, the system is determining which blocks in a stripe are free blocks. If writing to a new area of disk, then all the blocks are free, but if writing where a previously stored stripe is located then need to determine which blocks are free blocks (empty). Then when the number of free blocks is determined, one block is reserved for single disk failure recovery or two blocks are reserved for double disk failure recovery, and so on. The location of the blocks for redundant information are not reserved just the number of blocks. Then the data is assigned to the free blocks leaving any one or two blocks free. Then the redundant information is calculated and stored in the number of free blocks reserved. Then the data and redundant information are stored to the disks.

Accordingly, claims 1-12 and 14-44 are believed to be allowable over the §112 rejection.

Claim Rejections – 35 USC § 102

At paragraph 4 of the Office Action, claims 1-5, 22-25, 28-33 were rejected under 35 U.S.C. §102 as being anticipated by Patterson et al, “A case for Redundant Arrays of Inexpensive Disks (RAID), June 1988, hereinafter Patterson.

The present invention, as set forth in representative claim 1, comprises in part:

> 1. A system adapted to distribute redundant information across disks of an array, the system comprising:
>
> a storage operating system configured to invoke storage operations executed by a storage system, the storage operating system further configured to manage storage of information, including the redundant information and data, on blocks of the disks in response to disk access operations, the storage operating system including a storage module adapted to compute the redundant information in response to a layout of the data in stripes across the disks, ***the storage operating system maintaining at least one unallocated block per stripe for use by the storage module to store the computed redundant information, wherein the at least one unallocated block used to store the redundant information is located in any disk and wherein the location of the at least one unallocated block used to store the redundant information is dynamically allocated in a non-fixed pattern by the storage module before each write request for each stripe.***

By way of background, Patterson describes a fifth level RAID system. RAID level 5 assigns parity in a fixed pattern across the disks usually a rotating pattern. One problem with RAID 5 is that it is hard to add new disks because of the fixed parity placement.

Applicant respectfully urges that Patterson does not describe Applicant's claimed novel ***the storage operating system maintaining at least one unallocated block per stripe for use by the storage module to store the computed redundant information, wherein the at least one unallocated block used to store the redundant information is located in any disk and wherein the location of the at least one unallocated block used to store the redundant information is dynamically allocated <u>in a non-fixed pattern</u> by the storage module before each write request is completed for each stripe.*** In further detail, Applicant's invention allows ***the at least one unallocated block used to store the redundant information is dynamically allocated in a non-fixed pattern by the storage module before each write request is completed for each stripe.*** Before each stripe is written across each disk, the storage module reserves at least one unallocated block to use for redundant information. Each unallocated block is selected from the one or more blocks that are not already allocated to data blocks in a stripe layout. The storage operating system then selects at least one unallocated block of the stripe to store redundant information. This allows for the redundant information to be placed in any order, varying from stripe to stripe. Additionally, dynamically allocating in a non-fixed pattern the redundant information allows for a new disk to be added and with a next stripe protecting the new disk with redundant information.

In sharp contrast, Patterson describes a RAID 5 system where parity information is stored in a rotating pattern across the disks changing after each stripe. Patterson does not disclose a non-fixed pattern. Additionally, there is no disclosure in Patterson of determining an unallocated block after data blocks are allocated and storing parity information in that block because Patterson selects the parity block first on a rotating basis through all disks, then stores the data blocks in the remaining blocks.

Accordingly, Applicant respectfully urges that Patterson is legally insufficient to anticipate the present claims under 35 U.S.C. §102 because of the absence of the Applicant's claimed novel ***the storage operating system maintaining at least one unallocated block per stripe for use by the storage module to store the computed redundant infor-***

mation, wherein the at least one unallocated block used to store the redundant information is located in any disk and wherein the location of the at least one unallocated block used to store the redundant information is dynamically allocated <u>in a non-fixed pattern</u> by the storage module before each write request is completed for each stripe.

At paragraph 5 of the Office Action, claims 36, 39, 42, and 44 were rejected under 35 U.S.C. §102, as being anticipated by Spiegel et al., US Patent No. 6,571,326, issued on May 27, 2003, hereinafter Spiegel.

The present invention, as set forth in representative claim 36, comprises in part:

> 36. A method for distributing redundant information across disks of an array with a plurality of blocks on each disk, comprising:
> determining which blocks are unallocated in a stripe across the disks;
> reserving unallocated blocks for storing the redundant information in one or more reserved unallocated blocks;
> arranging data in the stripe for the data to be stored in one or more allocated blocks across the disks of the array;
> ***assigning the redundant information in a non-fixed pattern to the one or more reserved unallocated blocks***; and
> ***writing the data in the allocated blocks and the redundant information in the one or more reserved unallocated blocks as the stripe across the disks of the array.***

By way of background, Spiegel discloses a machine readable medium to preallocate space for data. In particular a nonvolatile manager determines the amount of unallocated space on a nonvolatile memory. Then, a dry run reservation tracker is designed to copy contents of a logical block table buffer into a dry run buffer. If there is not sufficient unallocated blocks for the contents of the logical block table buffer, the request is modified. If there is sufficient unallocated blocks for the contents of the logical block table buffer, then the reservation tracker reserves the unallocated blocks for the logical table buffer. The system is used to determine the most efficient use of unallocated space for placing data in the nonvolatile memory. (Col. 5, lines 29-31).

Applicant respectfully urges that Spiegel does not disclose nor suggest Applicant's claimed novel ***assigning the redundant information in <u>a non-fixed pattern</u> to the one or more reserved unallocated blocks***, and ***writing the data in the allocated blocks and the redundant information in the one or more reserved unallocated blocks as the stripe across the disks of the array***. In further detail, Applicant's claimed invention is writing both data and redundant information to the stripe, where the data is first allocated to the blocks of the stripe layout. Then, the system determines which blocks in the stripe layout are not being used by data. One or more of the blocks unallocated for data are then reserved for redundant information, i.e., parity blocks.

There is no disclosure in Spiegel of ***assigning the redundant information in a non-fixed pattern to the one or more reserved unallocated block***, because Spiegel does not disclose anything on parity or redundant information.

Accordingly, Applicant respectfully urges that Spiegel is legally insufficient to anticipate the present claims under 35 U.S.C. §102 because of the absence of the Applicant's claimed novel ***assigning the redundant information in <u>a non-fixed pattern</u> to the one or more reserved unallocated blocks***, and ***writing the data in the allocated blocks***

and the redundant information in the one or more reserved unallocated blocks as the stripe across the disks of the array.

Claim Rejections – 35 USC § 103

At paragraph 6 of the first Office Action, claims 6-8, 11, 12, and 26 were rejected under 35 U.S.C. §103 as being unpatentable over Patterson, in view of Baylor et al, US Patent No. 5,862,158, hereinafter Baylor.

At paragraph 7 of the first Office Action, claims 14, 15, 18-20, 27, and 34-35 were rejected under 35 U.S.C. §103 as being unpatentable over Patterson, in view of Baylor, and in further view of Stallmo et al., US Patent No. 5,657,468, hereinafter Stallmo.

At paragraph 8 of the first Office Action, claims 9 and 10 were rejected under 35 U.S.C. §103 as being unpatentable over Patterson, in view of Baylor, and in further view of Ulrich, US Patent Application publication No. 2002/0124137, hereinafter Ulrich.

At paragraph 9 of this Office Action, claims 16, 17, and 21 were rejected under 35 U.S.C. §103 as being unpatentable over Patterson, in view of Baylor, and in further view of Stallmo, and in further view of Wiencko et al., US Patent No. 6,557,123, hereinafter Wiencko.

At paragraph 10 of this Office Action, claims 37, 38, 40, 41, and 43 were rejected under 35 U.S.C. §103 as being unpatentable over Spiegel, in view of Patterson.

Applicant respectfully notes that claims 6-12, 14-21, 26, 27, 34, 35, 37, 38, 40, 41, and 43 are dependent claims that depend from independent claims believed to be in condition for allowance. Accordingly, claims 6-12, 14-21, 26, 27, 34, 35, 37, 38, 40, 41, and 43 are believed to be in condition for allowance.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to depend from allowable independent claims.

The Applicant respectfully solicits favorable action.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Shannen C. Delaney/
Shannen C. Delaney
Reg. No. 51,605
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500